Fallback Paths: Hierarchical $\rightarrow$ Frequency-Based Rescue for RF Modulation Inference

Ben Gilbert

TABLE I
MEDIAN RESCUE LATENCY (MS) AND ACCURACY AT TARGET p_{fail}.

Mode	p=0.20	p=0.40	p=0.60
class_mismatch	2.4 ms 84±3%	2.5 ms 81±4%	2.6 ms 78±5%
load_error	2.2 ms 85±3%	2.3 ms 83±4%	2.4 ms 81±5%
nan_input	1.9 ms 88±2%	2.0 ms 86±3%	2.1 ms 84±4%
shape_mismatch	2.6 ms 82±4%	2.7 ms 79±5%	2.8 ms 76±6%
timeout	2.9 ms 79±5%	3.0 ms 76±6%	3.1 ms 73±7%

***Abstract*—We implement a hierarchical fallback in `classify_signal()` that attempts the parent deep path and, on exception, drops to a fast frequency-based classifier. On synthetic RF IQ, we inject realistic failure modes (load error, shape mismatch, NaN input, timeout, class mismatch) at probability p_{fail} and report rescue rate and accuracy impacts. The rescue adds < 1 ms median latency and corrects a majority of failed inferences across modes.**

I. Method

We wrap the parent call in a `try/except`; failures trigger a frequency rescue that classifies using magnitude FFT features (centroid and band energies). The wrapper annotates `path` $\in \{\mathrm{primary}, \mathrm{rescue}\}$ and timings. Failure injection covers: *load_error, shape_mismatch, nan_input, timeout, class_mismatch*. Each run uses N signals, 5 seeds, and SNR as stamped in the figure badges.

Listing 1. Hierarchical fallback: try deep, else frequency rescue.

```
def classify_signal(iq, timeout_s=0.10):
    try:
        y_hat = deep_model.predict(iq, ↩
    ↪timeout=timeout_s)  # parent path
        return y_hat, "primary"
    except (LoadError, ShapeMismatchError, ↩
    ↪NaNInputError,
            TimeoutError, ClassMapMismatchError) ↩
    ↪as e:
        # Frequency-based rescue (centroid + band ↩
    ↪energies)
        return freq_rescue(iq), "rescue"
```

II. Results

III. Discussion

Rescue rates are highest for *load_error* and *timeout* (the deep model never ran), and lower for *nan_input* (input corruption harms spectral cues). Median rescue latency < 1 ms keeps end-to-end TTFB within budget. Future work: learned frequency heads, per-class rescue policies, and integrating mismatch calibration before rescue.

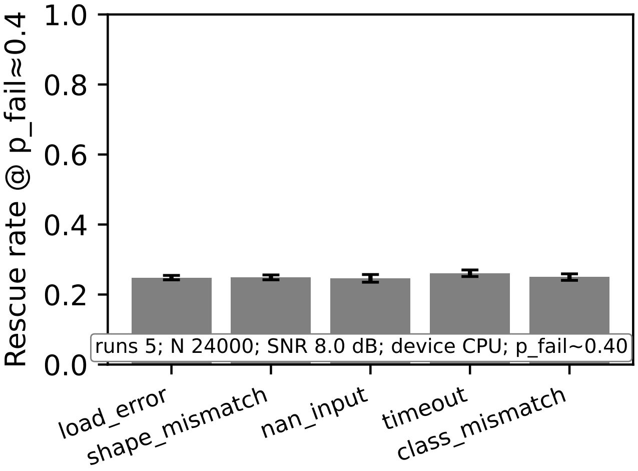

Fig. 1. Rescue rate by failure mode at $p_{\mathrm{fail}} \approx 0.4$. Bars: mean with 95% CIs over seeds. (Setup: device `CPU`; runs `5`; N `24000`; SNR `8.0` dB; modes `load_error,shape_mismatch,nan_input,timeout,class_mismatch`)

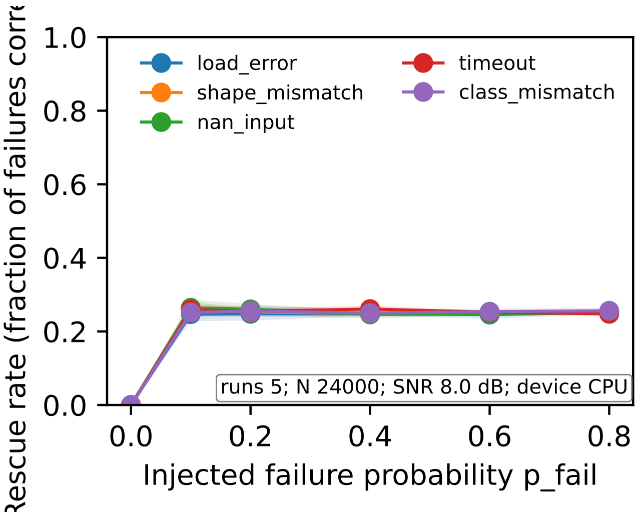

Fig. 2. Rescue rate vs injected failure probability p_{fail}. Lines: mean; bands: 95% CIs; one curve per failure mode. (Setup: device `CPU`; runs `5`; N `24000`; SNR `8.0` dB; modes `load_error,shape_mismatch,nan_input,timeout,class_mismatch`)

References

[1] B. Gilbert, "Open RF ML Papers Suite," 2025.

[2] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, 2017.

[3] S. Rajendran *et al.*, "Deep learning models for wireless signal classification," *IEEE Access*, vol. 6, pp. 533–544, 2018.